1

TR-013

BM/C^3 FORCE MODEL

VLSI/VHSIC DIGITAL PROCESSING:
A COST METHODOLOGY

D. C. MORRISON

OCTOBER 1988

Prepared for

ELECTRONIC SYSTEMS DIVISION (ESD/ACC)
HANSCOM AFB, MA 01731-5000
CONTRACT NUMBER F19628-88-D-0007
TASK ORDER 0001

TECOLOTE RESEARCH, INC.
5266 HOLLISTER AVENUE, NO. 301
SANTA BARBARA, CALIFORNIA 93111
(805) 964-6963

ACKNOWLEDGMENTS

A person working alone learns very little. It is through contact with others that we learn the most. In recognizing this, I would like to thank Richard Elmhurst of Honeywell (Clearwater) and Jerry Moylan of Honeywell (Plymouth) for arranging useful and informative plant visits. I also thank Doug Theis of the Aerospace Corporation for taking time out of his busy schedule to field my numerous phone calls. Lastly, I would like to thank Arve Sjovold of Tecolote Research, whose constructive criticism and intelligent comments were very much appreciated.

CONTENTS

FOREWORD

The primary objective of the paper is to investigate estimating methodologies for special-purpose, mobile-based Digital Processors with VLSI/VHSIC technologies.

This paper also documents the First Order* Processor Estimator (FOPE) for processors based on ground, mobile, airborne, or spaceborne platforms. In addition, this paper will serve as a primer on processing cost drivers and major technical issues. Much of the information contained in this report has been collected from informal sources (i.e., telephone conversations, sales brochures, catalogs, etc.), and thus is presented fairly informally. No attempt is made to collect proprietary or sensitive information, or to investigate or present exotic or esoteric technical issues regarding processors. The information is presented using lay language. Technical jargon is avoided and concepts are kept simple. The idea is not to document a rigorous engineering breakthrough, but rather to discuss relevant technical and cost issues regarding special-purpose digital processors. Appendix A contains a short discussion of off-the-shelf general-purpose processing suites.

*First Order Approximation

1
BACKGROUND

The FOPE model emerged from the need to estimate the cost of special-purpose processing equipment that is contained in mobile platforms, especially those based in space. Analysts for the Strategic Defense System (SDS) are keenly interested in the technical feasibility of the space-based processing requirements of SDS and subsequent processor cost given such requirements. Processor requirements for the SDS are given in terms of throughput (performance in terms of millions of floating point operations per second and millions of instructions per second) and memory. Cost estimating relationships or methodologies that use these requirements as cost drivers are certainly desirable.

Historically, the cost of special-purpose digital processors has generally been estimated using power, weight, and/or volume as cost drivers. SDS processor power, weight, and size specifications (when known) represent constraints and not necessarily the actual performance of the processor. The sizing of SDS processors have primarily concentrated or processor throughput and memory size. Therefore, this model's methodology departs from the traditional physical cost drivers and uses throughput and memory as drivers in an engineering build-up approach.

The term "special purpose processing" means (for thi report) mobile-based processing equipment that is not commercially available and processing suites that are constructed with designs carefully tailored to specific applications. It is possible to postulate that commercially available data processing suites can be loaded into trucks, airplanes, etc., and be used to fulfill processing requirements. For this type of application, the processor might be called a "general-purpose processor" (i.e., commercially available off the shelf, little or no retrofitting required). In this case, off-the-shelf procurement costs are known and retrofit costs, if any, are usually minimal.

The requirement for special-purpose processing is born from the fact that most often off-the-shelf processors cannot be used in hostile environments, and also because off-the-shelf processing will not accommodate the specialized or customized processing that is prevalent in many of today's military applications. Special-purpose processors are usually constrained by physical requirements such as power, weight, and size. Power, weight, and size constraints are primarily driven by the type of basing mode. This report considers only mobile basing modes, either ground, airborne, or space. Special-purpose processing, then, can be characterized in the following way:

1. Not commercially available
2. Mobile basing mode
3. Physical constraints
4. Specialized processing requirements
5. Hostile operating environments
6. Unusual reliability requirements

It is clear all six characterizations add significant cost over and above what is expected for general-purpose processors. These characterizations also determine the type of technology employed.

2
METHODOLOGY

The dearth of cost and technical data for processors, particularly for the space-based processors that employ VLSI and VHSIC technologies, dictated that an engineering build-up methodology be used. After a review of special-purpose processors that are currently being employed (ground mobile, airborne, and space), it became obvious that a rich data base that encompassed SDS processing requirements did not exist. In fact, most of today's operational military electronics in trucks, aircraft, and satellite delivery systems (Shuttle, Titan, etc.) use processors whose technology is proven and at least 5-10 years old. Given this fact, a cost estimating relationship (CER) methodology was simply out of the question, and an engineering build-up method for costing special-purpose processors was selected.

The engineering build-up methodology allows for a certain amount of flexibility. Since costs are built up from H/W componentry (integrated circuits), the variation of processor costs due to basing mode is simply handled by inputting H/W costs that reflect the environment that the H/W is likely to see. This idea makes good sense when one considers how processing chips are priced. The major cost drivers (supply and demand effects are ignored) appear to be chip operating temperature, chip speed, chip hardness, chip reliability, and chip density (i.e., type of integration technology). The type of basing mode or operating environment is a major factor that determines the requirements for chip operating temperature, speed, hardness, and reliability. Therefore, all things being equal, if two chips are compared, each being functionally equivalent, the cost difference can be attributed to the difference in basing mode or application. In reality, however, the delta cost is usually tainted by economics external to the chip (i.e., pricing strategies). No attempt is made in this study to deduce supply and demand related pricing strategies.

Another area of model flexibility has to do with technology. Advances in technology are captured by increasing the performance (throughput) of each chip or chip set. This parameter effectively reduces the number of chips required per function. If prices per chip have not increased commensurately, then cost per function should decline. This reflects the promises of technology advance.

The central idea of a build-up method to cost special purpose processors lies in the fact that the processor can be broken down into generic functional elements. This study uses the following functional elements:

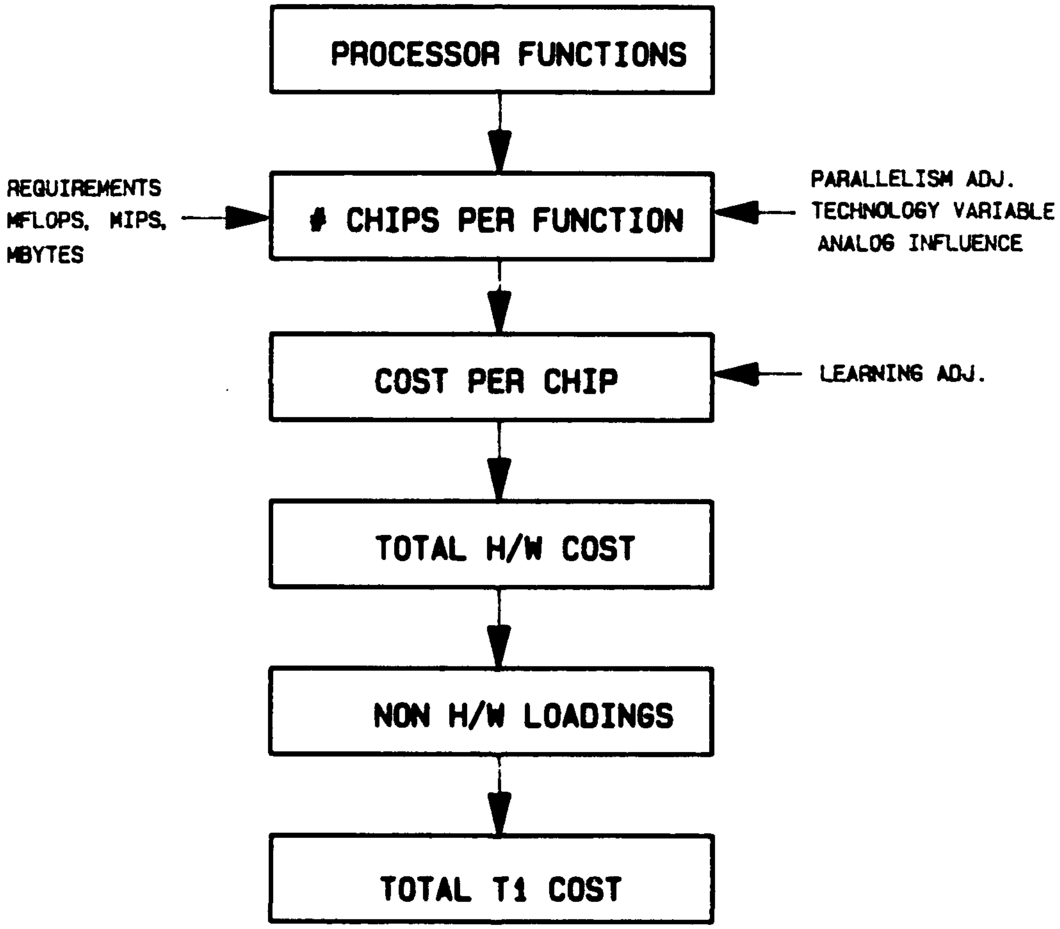
		Generic Functional Elements	FOPE Equivalent	Measure of Performance
Processing System	CPU	1. Control Unit	Scalar Processing	MIPS
		2. Arithmetic-Logical Unit	Scalar/Vector Processing	MIPS/MFLOPS
		3. Main Memory	SRAMS, DRAMS, PROM	Megabytes
		4. Mass Memory	Bubble, Disk	Megabytes
		5. I/O Interface	Other	--

Measure of performance can then be used to describe each functional element. The I/O interface (buses, interface chips, etc.) is a notable exception. The FOPE model assumes that I/O interface chip costs are driven by CPU chip costs. Therefore, the I/O interface function has no measure-of-performance parameter assigned to it.

The measure-of-performance variable determines the number of chips required to achieve the desired performance for each functional element of the processor. The processor architecture and performance per chip ratio is based on two developmental VHSIC processors. One is the space-based Generic VHSIC Spaceborne Computer (GVSC), and the other is the airborne-based Common Signal Processor (CSP). If desired, the ratio can be changed to reflect other analogs. Once the number of chips are calculated, it is a simple matter to multiply by cumulative average chip cost to obtain total H/W cost for each functional element. Non-hardware loadings such as material handling, factory overhead, touch labor, G&A, and fee are factored from total hardware costs. Enclosure costs can be added (if desired) to complete the total box cost. Full documentation and use of the model appears in section 5.

In summary, the model employs an engineering build-up methodology with analogs. A generic processor architecture is broken down into five functional elements. The hardware cost for each element varies depending on basing mode and desired performance. Technology advance variables can also be used to modify hardware costs. Finally, factored non-hardware costs are added to complete the total processor cost.

FIRST ORDER PROCESSOR ESTIMATOR



3
FUNDAMENTAL PROCESSING DEFINITIONS/ISSUES

Before discussion of the FOPE model, it is recommended that the reader familiarize himself with some basic definitions that pertain to digital processors and their corresponding technologies as presented in this section. Also, this section can be used later as a quick reference that helps to clarify and define concepts and issues that are important when one embarks on the costing of VHSIC-based processors.

This "definitional" section is broken into three major subsections:

1. Basic Processor and IC Technology Definitions
2. Processor Functions
3. Parallel Processing

The definitions for the first two sections are for the most part extracted from From Chips to Systems by Rodnay Zaks and Alexander Wolfe. It is an excellent, well-written, and organized discussion of processor components to full-up processing systems. The interested reader should consult this book if a more detailed understanding of processing architectures is needed. The parallel processing section definitions and ideas come mainly from Principles of Parallel and Multiprocessing by George Desrochers. There were other sources used for the compilation of the definitions. Please consult the bibliography for a complete listing of sources.

3.1 Basic Processor Definitions

Integrated circuit (IC) - A device that integrates a circuit of several electronic components in a single package.

Integration Level - Designates the type of technology that integrates transistors on a single integrated circuit.

a. Small Scale Integration (SSI) 1 → 10 transistors per chip

b. Medium Scale Integration (MSI) 10 → 250 transistors per chip

c. Large Scale Integration (LSI) 250 → 15,000 transistors per chip

d. Very Large Scale Integration (VLSI) 15,000 → 400,000+ transistors per chip

e. Very High Speed Integrated Circuit (VHSIC) - VLSI technology and a speed requirement of 25 mHz or 40 nanoseconds cycle time

Transistor - A tiny chip of crystalline semiconductor material, usually silicon, that amplifies or switches electric current.

Gate - Incorporation of several transistors, which implements logical functions such as AND, OR, and NOR.

Packaging - The physical process of locating, connecting, and protecting integrated circuit. Types of packaging include:

1. Plastic Dual In Line Package (DIP)
2. Ceramic DIP
3. Thin DIP
4. Leadless Chip Carrier
5. Plastic Leaded Chip Carrier
6. Pin Grid Array
7. Flat-Pack
8. Multichip Carrier

Access Time - A time interval that is characteristic of a storage device. Essentially, it is the time between the application of a specified input pulse (assuming that other necessary inputs are also present), and the availability of valid data signals at an output.

Cycle Time - The total time required by a device to complete its cycle and become available again. One cannot assume that the processor with the fastest cycle time will be the best overall performer in a particular application. Other parameters that influence a processor's performance include the flexibility and power of its instruction repertoire and the number of storage cycles it requires to execute each instruction. Typically, the access time will be shorter than the cycle time, though they may sometimes be equal.

Switching Speed - The time it takes for a transistor to change states (cutoff to saturation or vice versa) upon sensing the appropriate signal. Also the signal (pulse) has to be of sufficient length so the state change can take place. Required pulse duration is what limits maximum gate frequency.

Wafer - Slice of semiconductor crystal material used as a substrate for ICs, diodes, and transistors.

Die - Synonymously called a chip. A tiny piece of semiconductor material, cut from a wafer, on which one or more active electronic components are formed to comprise an integrated circuit (on the order of 5 x 5 mm).

MIPS - Millions of instructions per second. Historically, MIPS have been calculated by taking the elapsed times for each instruction (load, add, multiply, etc.) and applying them to a typical percentage mix of instructions. MIPS, therefore, are calculated before any actual benchmarking on the computer is done.

MFLOPS[*] - Millions of floating point operations per second. Historically, MFLOPS have been calculated by exercising a benchmarking program. A benchmarking program is typically written in a high-level language, and the frequency and type of statements that appear reflect the characteristics of the intended application. Because benchmarking programs are typically written in high-level language, they are as much a test of the efficiency of compiler-generated code (overhead included) as well as the system upon which they are executed.

The historical definitions of MFLOPS and MIPS have given way to new meanings, however. Because the marketing of today's processors is heavily dependent on favorable performance ratings, the definition and meaning of MFLOPS

[*] Millions of operations per second (MOPS) may be used interchangeably with MFLOPS, depending on whether floating point or fixed point calculations are being done.

and MIPS have been clouded. Processor makers now use benchmarking to measure both MIPS and MFLOPS. Not only is there no clean conversion from MIPS to MFLOPS, but there is no guarantee that a MFLOP = MFLOP = MFLOP because of the fact that benchmarking is application-dependent (i.e., the performance measure depends on application and subsequent processor architecture).

This paper tries to clarify the MFLOP vs MIP issue by assigning each performance measuring variable to a particular function of the processor. Since the decoding of instructions is especially prevalent in logic and control functions of the processor, MIPS is assumed to be a measure of the performance of the control and logic functions. Typically, program flow control instructions such as conditional branches, Boolean operations, and other special-purpose instructions are processed most efficiently in a sequential manner (i.e., scalar processing). This type of processing is usually implemented by what is known as a Von Neumann architecture. Situations in which programs can alter themselves (modification of instruction streams) are a characterizing feature of a Von Neumann architecture.

The MFLOP performance variable is reserved for the description of the arithmetic function of the processor. Multiply/add and subtract/divide operations are usually implemented by highly parallel or multiprocessing architectures where branching and logic are minimal. One type of parallel architecture that might be implemented is called Vector parallelism. Vector parallelism is distinguished by the fact that the same operation is performed on all the elements of the vector at once. To be efficient, the vector must be of sufficient length (> 10 elements) and a maximum number of the elements in each vector must be exercised. For example, with a vector processor comprised of n processing elements, vectors with up to n elements can be processed efficiently. For operations on vectors with less than n elements, some of the processing elements are not exercised. Thus, if a vector processor was given a vector of short enough length, its performance would be simply that of a scalar machine. To counteract the inefficiency of short vector lengths, a scalar unit is usually added to the vector processor to serve as program control and operation of the vector processor.

IC technologies are varied, and many are simply variants of a single fundamental technology. The following discussion documents several of the most prevalent, including MOS, Bipolar, and Gallium Arsenide.

MOS (Metal Oxide Semiconductor) technology is the technology used to create transistors and other components on the surface of the small piece of silicon. There are many ways MOS technologies are implemented; the following text discusses the notables.

A. PMOS Technology. PMOS is a relatively old MOS technology that is well understood and economical. It was used extensively for manufacturing all of the first microprocessors. PMOS technology gives excellent density (up to 20,000 transistors or more per chip). However, it is relatively slow when compared with newer technologies such as NMOS.

B. NMOS Technology. NMOS is intrinsically faster than PMOS, as it uses electrons rather than holes as charge-carriers. It gives an excellent density and is generally regarded as the best compromise for implementing fast and complex microprocessors today. Being a newer technology, however, it is not as well developed as PMOS, and it is not used by all manufacturers. It is typically at least twice as fast as an equivalent PMOS microprocessor.

C. CMOS (Complementary MOS) Technology. The characteristics of CMOS technology lie somewhere between those of the NMOS and PMOS technologies. CMOS is faster than PMOS, but it is somewhat slower than NMOS and achieves a good density. Because it uses two transistors rather than one, however, CMOS offers less integration than NMOS.

Essential advantages of CMOS are that it has very low power consumption (will operate between 2V and 12V) and has an excellent noise immunity (40%, which is almost ideal). CMOS technology was specifically created for avionics and aerospace applications.

D. Bipolar Technologies. Bipolar technology is one of the fastest technologies available today. Within the various bipolar technologies, the main technology is low-power-Schottky TTL (LPSTTL). It is used for the implementation of fast bit-slice devices. LPSTTL results in instruction execution times of 70 to 100 ns per instruction versus approximately 1 μs (microsecond) for a monolithic microprocessor. The two main disadvantages of bipolar technology are:

1. high power dissipation
2. low density

E. Gallium Arsenide (GaAs). Unlike silicon, gallium arsenide does not exist in nature and must therefore be synthesized. The technology of GaAs has developed very rapidly in the past decade and 3-inch diameter substrates of much improved quality are now available for digital, analog, and optical applications.

Digital GaAs ICs are characterized by high speed, low power (100 pico sec gate delays at .15 mW), radiation hard (10 megarad total dose), low density (SSI and MSI applications to date only), and high expense. Digital GaAs technology is not nearly as mature as silicon technology. This fact has made it difficult for GaAs to assimilate with the other mainstream (MOS, Bipolar) silicon technologies.

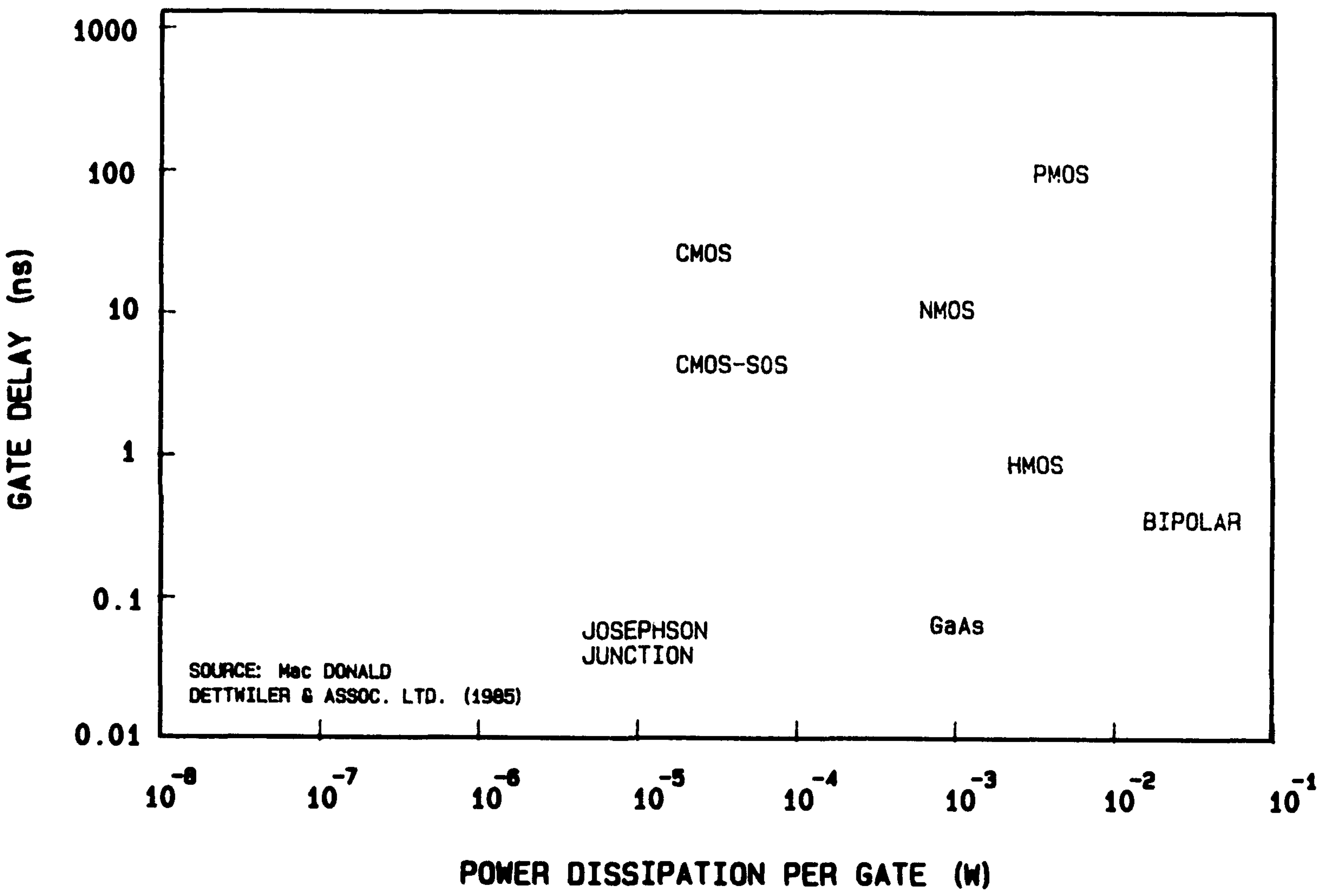
Figure 3.1 shows how different technologies compare in relation to power and speed.

3.2 Processor Functions

Figures 3.2, 3.3, and 3.4 show the functions that are germane to every processor system: Control, ALU, Memory, Input, and Output. These functions are provided by specific components that can be grouped by function. The three basic system components are: CPU, Memory, and I/O Interface Chips.

Central Processing Unit:

A. **Arithmetic/Logical Unit (ALU)** - Performs arithmetic and logical operations on the data passing through it. Typical arithmetic



SOURCE: Mac DONALD
DETTWILER & ASSOC. LTD. (1985)

Figure 3.1. Performance summary of selected digital IC technologies.

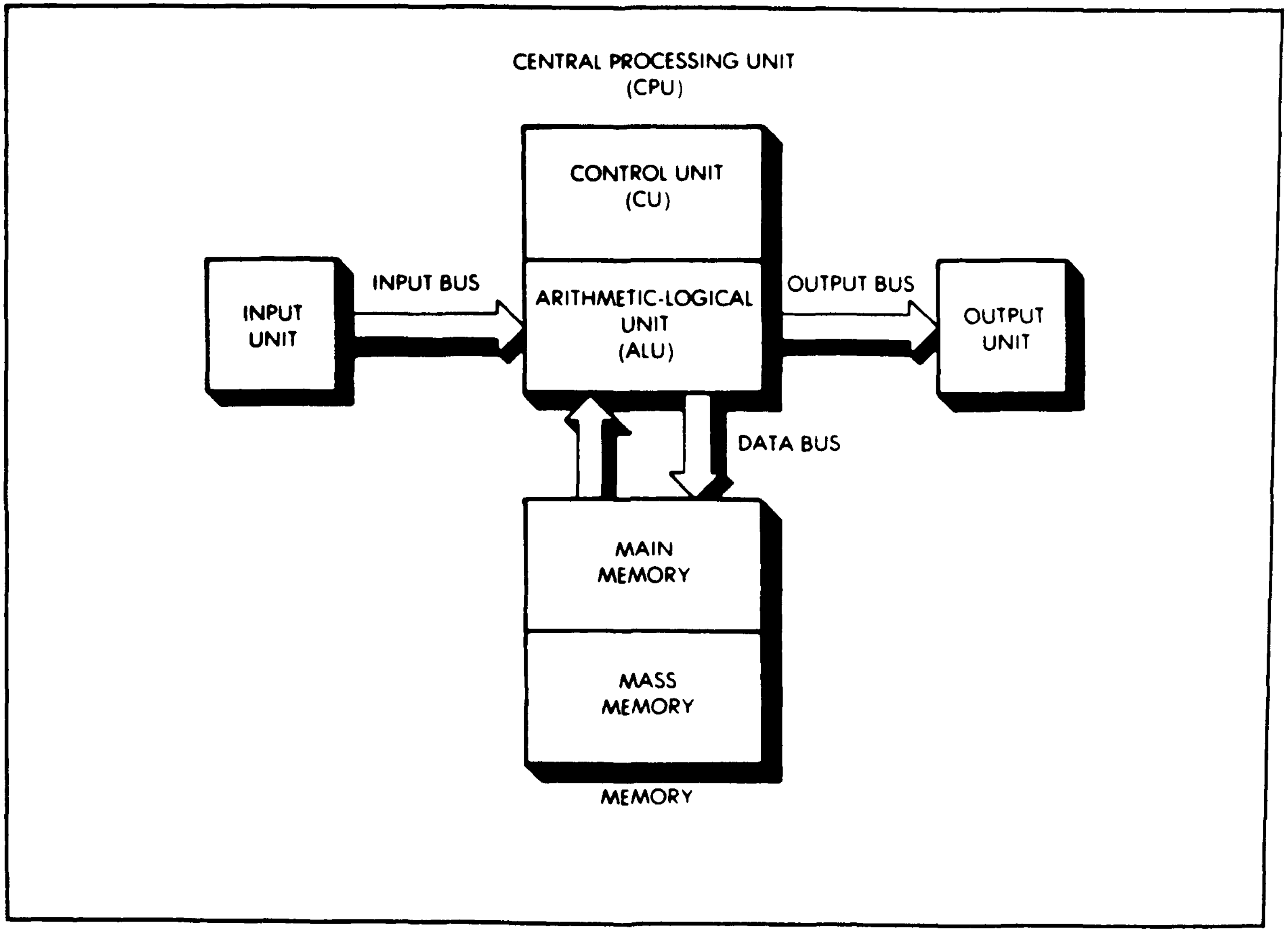


Figure 3.2. The five functional units of a computer system.

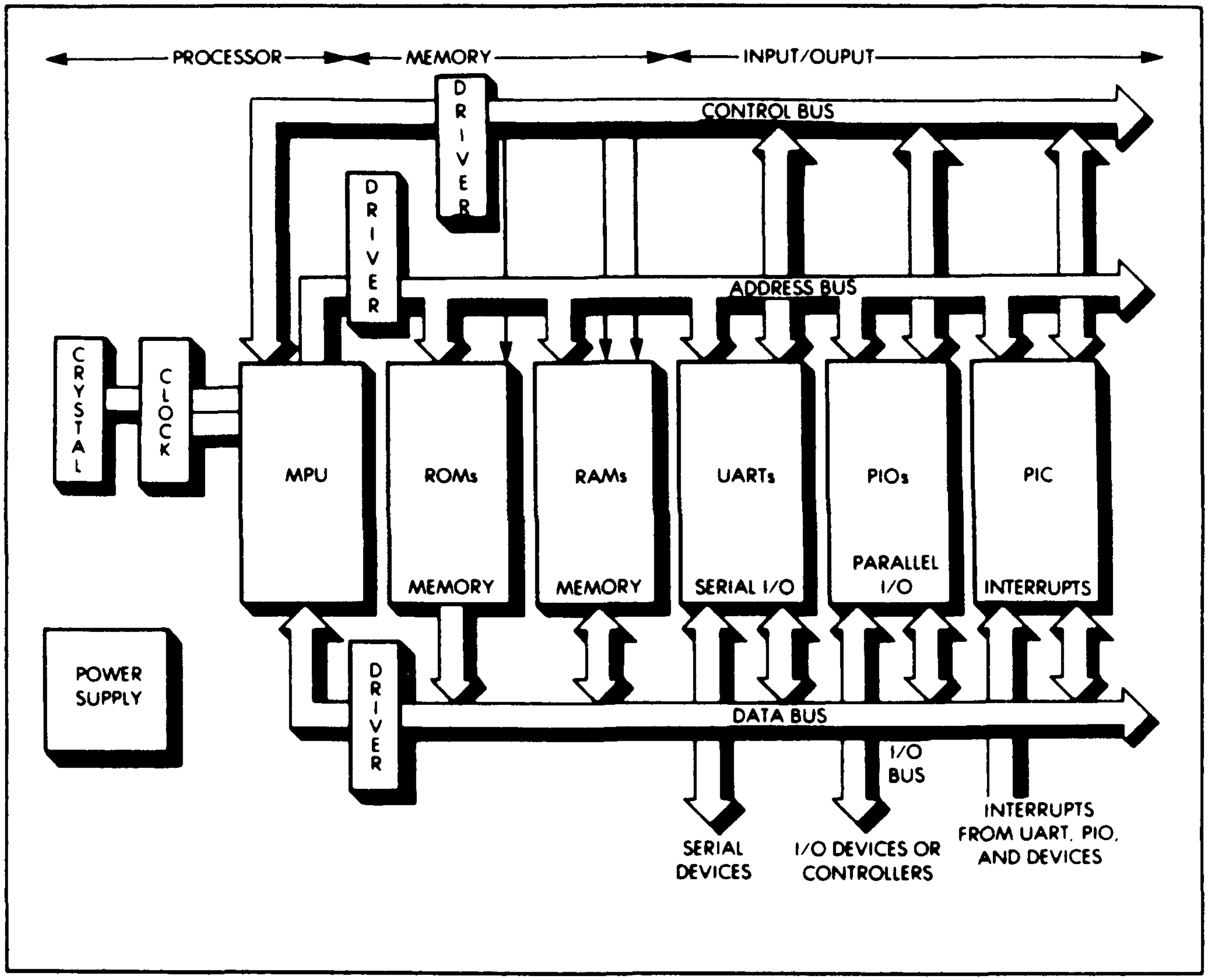


Figure 3.3. A standard microprocessor system.

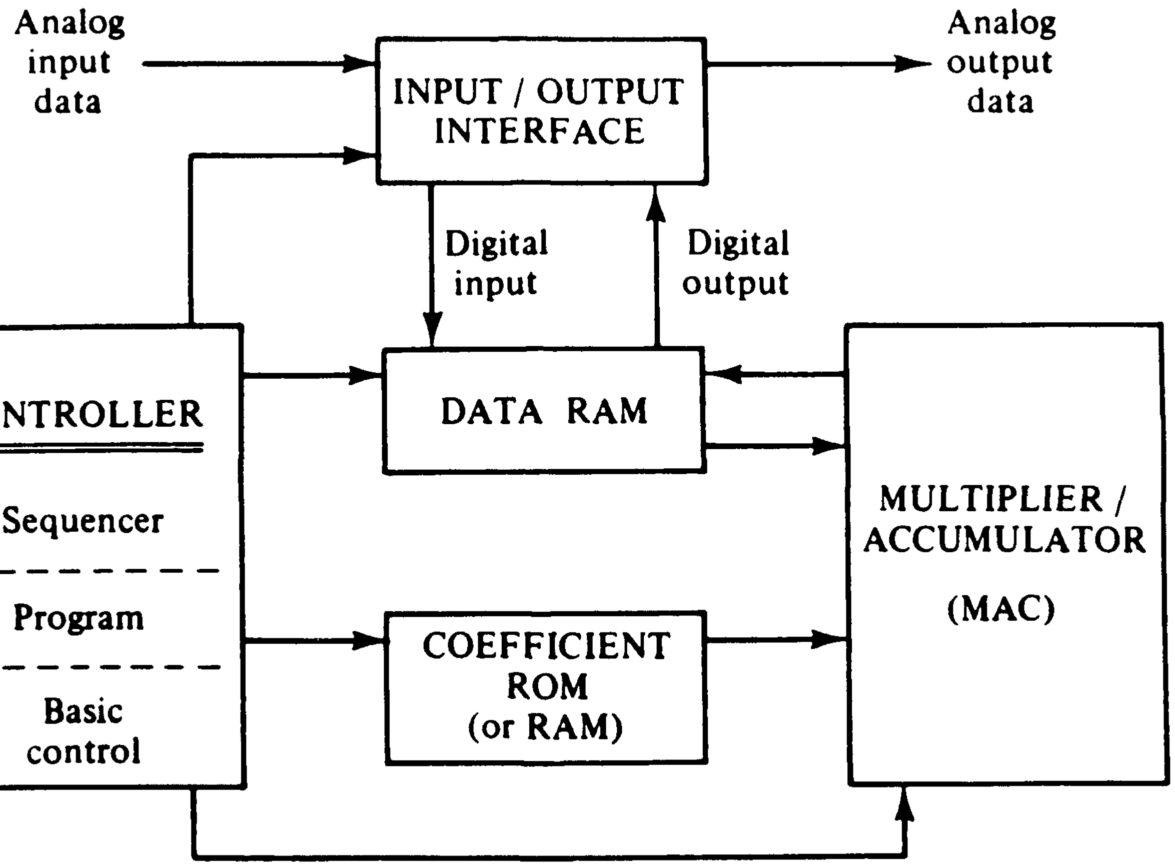


Figure 3.4. General structure of a digital signal processor.

functions include addition and subtraction. Typical logical operations include logical AND, logical OR, and shift operations.

B. **Internal Registers** - Being part of the ALU, they provide the fastest level of data memory available to the system. Typically, the contents of internal registers can be accessed by the ALU in less than 100 ns. Usually there are few internal registers.

C. **Control Unit** - The main functions of the control unit are to fetch, decode, and execute the successive instructions of a program stored in the memory. The Control Unit sequences the operation of the entire system. In particular, it generates and manages the control signals necessary to synchronize operations, as well as the flow of program instructions and data within and outside the ALU. The CU controls the flow of information on the address and data bases, and interprets and manages the signals presented on the control bus. The control unit of most processors is implemented with a microprogram. Read only memory (ROM) stores the microprogram and generates the sequence of micro-operations required to fetch and execute external instructions.

Memory:

A. **Cache Memory** - A high-speed storage unit that can significantly increase the performance of a computer by serving as a fast-access buffer between main storage and the central processor on the input/output subsystem.

B. **Main Memory** - Usually called the "memory" of the system. It is implemented in one or more components, depending on its size. Main memory is characterized by being fast (20 → 200 ns) and expensive. Main memory is usually implemented using random access memories (RAM), read only memories (ROM), or a combination of both. RAM refers to the fact that any of its contents may be accessed at any time. This is in contrast to a serial memory, such as a magnetic tape, where access to the stored data is only in a fixed order, e.g., the serial order in which data passes under the tape head. Actually, both ROM and RAM are random access memories, but the term "random access" traditionally refers to read/write memories and is, therefore, only used with RAM. Two technologies are used for RAM memories: static and dynamic.

A static RAM stores a bit of information within a flip-flop. The flip-flop is typically implemented with 4 or 6 transistors. It is asynchronous and does not require a clock. The contents of a static RAM remain stable forever, as long as power is available.

A dynamic RAM stores a bit of information as a charge. The obvious advantage of a dynamic RAM is that this elementary cell is smaller (only one transistor) than a static RAM flip-flop, resulting in a much higher density. For example, a 64K-bit dynamic RAM resides on the same chip area as a 16K-bit static RAM.

The disadvantage of a dynamic RAM is the increased complexity of the memory board due to the necessity of additional logic. Like any charge, the charge stored in the capacitor leaks, and within a few milliseconds most of the charge is lost. To preserve the information contained in a dynamic RAM memory, the charge must be refreshed every 1 or 2 milliseconds.

The main disadvantage of RAM is volatility. Whenever power is removed, the contents of RAM are lost. That is why ROMs are preferred for storing programs that would otherwise need to be reloaded too frequently.

Conversely, under varying external conditions, such as the time of day, different programs may have to reside at the same address in memory, which is not possible with ROM. If ROM is used to provide nonvolatile memory, then all the programs will have to reside at different addresses simultaneously (i.e., end to end) within the ROM, and this requires a very large ROM memory.

Recall that a read only memory is a memory whose contents, once written, can only be read. Writing data in a ROM is generally called programming the ROM, since a program is usually what is written. However, programming here means that the specified bit patterns have been written into a memory. Because a read only memory is intrinsically nonvolatile, it is almost always used to store control programs.

The three major disadvantages of a ROM are:

a) The delay involved in producing ROMs.

b) The large quantity that must be produced at one time.

c) The fact that, once manufactured, a ROM cannot be modified. In other words, should an error be found in the program, it is impossible to effect a change within the ROM: the ROM must be replaced.

These constraints would retard the development phase of a system and inhibit the production of systems in small numbers. For these reasons, several other types of read only memories have been introduced, all of which can be directly programmed by the user.

<u>EPROM</u> - Erasable Programmable Read Only Memories are read-only user programmable memories that can be reprogrammed a number of times. There are two main types: the ultraviolet (UV)-erasable PROM and the electronically erasable PROM. The main difference between the two is the way the memories are erased (zeroed). The UV EPROM is zeroed by exposing it to hard ultraviolet light; the EEPROM is zeroed electronically. Once zeroed, selected locations within the EPROM can be programmed by installing new bit patterns. EPROMS are fairly expensive (almost as expensive as static RAMS), but generally slower than static RAMS.

C. **Mass Memory** - Is used to provide low-cost, high-capacity storage. Special storage devices are used to store large quantities of data on inexpensive supports.

Bubble Memories - Bubble memories can provide a large amount of non-volatile memory that is immune to harsh operating environments such as vibration and radiation. They retain information for long periods of time. However, they are slow (millisecond range) and relatively expensive. They are generally used only in highly specialized applications such as military systems or other applications where rugged operating environments are present, and access time is not an issue.

Disk - Although there are many types of disk storage devices (floppy, hard, solid state, optical) representing differing performance levels, disk memory is generally the most inexpensive way to store non-volatile data. Access times are slow (millisecond range), but large amounts of data can be stored cheaply. High-performance disk memory used by many of today's supercomputers achieve high densities and faster speed (microsecond range) but cost significantly more. Cray's solid-state storage device is one example of this type of disk. Disk as a storage device usually has applications in a fixed ground-based environment. They also involve mechanical parts which may preclude use in some environments.

I/O and Interface Chips:

Three types of I/O circuits have been developed to facilitate the connection and management of I/O devices to a microprocessor system.

A. Interface Chips
B. Scheduling Chips
C. Device-Controller Chips

The basic interface chips are essentially passive devices, i.e., devices that do not execute any complex operation. The scheduling chips are devices that facilitate or implement a scheduling algorithm. The device controller chips provide the required interfacing and sequencing for the specific device they control.

As far as the FOPE model is concerned, I/O and interface chips, control, data, address buses, and drivers are considered to be other electronics.

3.3 Parallel Processing

Simply put, parallelism is the ability to do more than one activity at once. The study and design of parallelism in processing architectures was born from the fact that total processor performance requirements have far exceeded the performance of a single processing element. To achieve the throughput requirements, designers have resorted to architectures that interconnect and network individual processing elements efficiently.

The efficiency of parallelism not only depends on the hardware architecture, but also on the software and the problem to be solved. Software tools such as vectorizing compilers and multitasking languages (Ada) have helped exploit parallelism. Even if the hardware and software problems are solved, one must still ask, can the problem be split in n equal partitions to be worked on by n processing elements, and afterwards can the results be assembled back together to yield meaningful results?

The efficiency of parallelism can be measured by comparing the work done by one processor versus the work done by n processors over a given length of time. The ratio of the two is called "speedup factor." There are limitations to the speedup factor, however. Simply connecting huge arrays of n processing elements does not guarantee a speedup factor of n. As more and more processors are connected, more complex controllers are needed and larger, more complex operating systems are needed. In other words, the system's overhead functions (program control) are getting increasingly larger so that each additional processing element is adding less and less to the overall performance of the processor. This idea is very similar to the economic idea of diminishing returns. There are three conjectures that are presented in this section that deal with the problem of parallel processing speedup factors. They are ideal, Amdahl's conjecture, and rule of thumb ratio. Figure 3.5 shows the comparison between Amdahl and Ideal.

1. **Ideal** - This conjecture states that the speedup factor is linear (i.e., no diminishing returns). For example, if 10 3-MIP processing elements are connected, then total performance is 30 MIPS. For this to happen, the following must be true:

   A. The problem must be able to be broken into n independent equal parts and be reassembled into meaningful results.

   B. No additional or low overhead by larger architectures.

   C. No conflicts over memory access of communication paths.

*2. **Amdahl** - In reality, A, B, and C from above cause speedup to be less than ideal. Therefore, there is a penalty to be paid in the form of this speedup equation:

   $$\text{Speedup} = n/\ln n$$

   For example, instead of needing 10 3-MIP processing elements to achieve 30 MIPS, the Amdahl equation would state that 36 3-MIP processing elements are needed to achieve 30 MIPS.

   $$10 \approx 36/\ln 36$$

*3. **Overhead Ratio** - This rule of thumb simply states that for every 11 processing elements, one more processing element must be added just to cover overhead.

Parallelism Definitions

Scalar Processing - Processing that is done sequentially, one element of the vector at a time. Scalar processing is most effective for program control

*The FOPE model allows for either the Amdahl estimate or the overhead ratio.

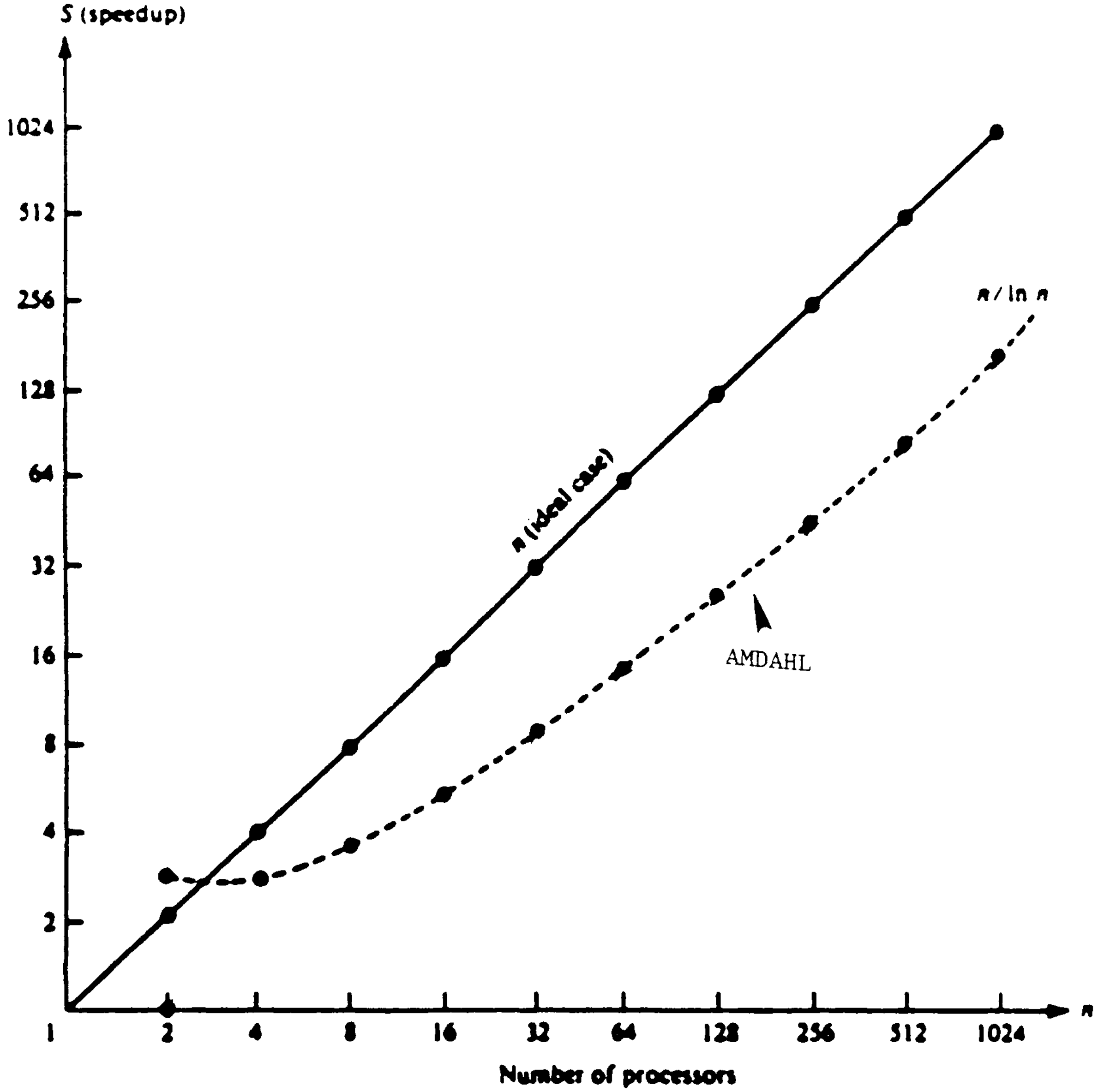


Figure 3.5. Various estimates of the speedup of an n-processor system over a single processor.

functions where there is conditioned branching, Boolean operations, and other special-purpose instructions. Sometimes scalar processing is called Single Data Single Instruction (SISD) processing. Von Neumann architectures are an example of this type of processing. See figure 3.6.

Vector Processing - Processing that is characterized by the performance of the same operation on all elements of the vector at once. On a vector machine with n processing elements, vectors with up to n elements can be processed efficiently. Larger vectors are broken into parts that can fit into the n element space, and the parts are processed serially through the vector unit. For operations on vectors with less than n elements, some of the processing elements are turned off and produce no result for that cycle. Note that if a vector processor is given sufficiently short vectors, then its performance will be similar to that of a scalar processor. This is the reason that many processors use both a scalar and vector processor. That way, conditional branching or Boolean instructions are never received by the vector processor, and are processed more efficiently by the scalar processor. The vector processor is then free to receive large vector lengths, as is the case with the arithmetic functions. The FOPE model takes exactly this view. Scalar processing is reserved for control functions and is measured in millions of instructions per second (MIPS), while the vector processing is reserved for arithmetic functions and is measured in millions of floating point operations per second (MFLOPS).

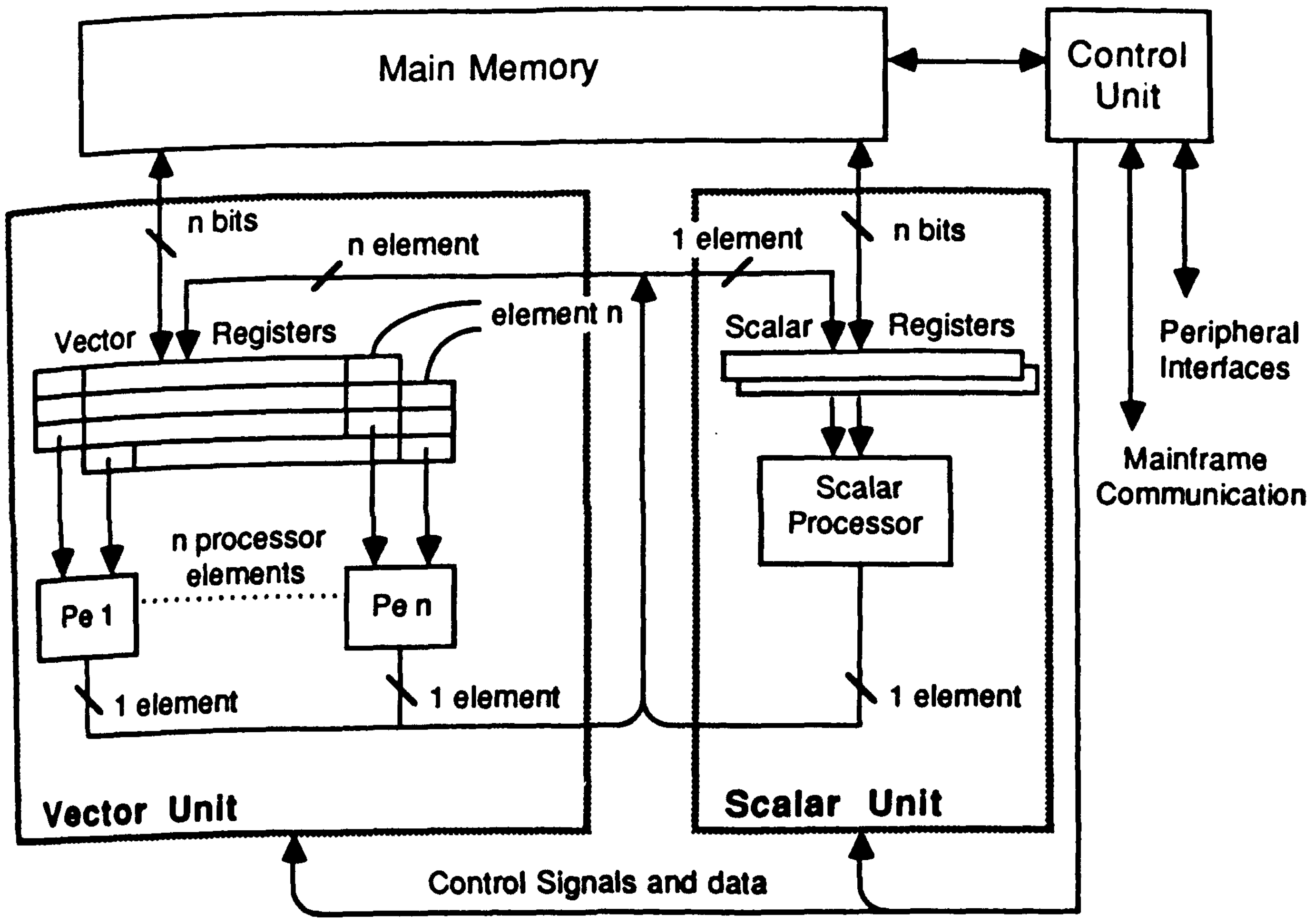


Figure 3.6. Vector processor architecture.

4
FIRST ORDER PROCESSING ESTIMATOR ANALOGS

One of the primary requirements of the FOPE model is the ability to cost processors based on the performance and type of processing. Total processing performance is measured using throughput reflected both in MFLOPS and MIPS. These throughput variables correspond to the type of processing that is being done. More specifically, MFLOPS is a measure of the amount of vector processing and MIPS is a measure of scalar processing. Please consult the definition section for a complete description of, and interrelationships between, MIPS, MFLOPS, Scalar, and Vector processing.

In order to capture both types of processing, two types of processors were investigated and used to calibrate the FOPE model. The vector processing selection was IBM's Common Signal Procesor (CSP). The scalar processor analog was based on Honeywell's Generic VHSIC Spaceborne Computer (GVSC). The following discussion is devoted to presenting the background and technical details of each processor. The discussions, for the most part, are synopses of the marketing brochures for each machine.

Generic VHSIC Spaceborne Computer

At the heart of the GVSC are three 1750A processors. Each 1750A processor consists of three CPU and two floating point processor chips (FPP). This architecture is estimated to deliver 12 MIPS of throughput in worst case environments. The CVSC architecture is expected to consume 80 watts (nominal) and weigh 12 pounds. It consists of six double-sided, printed circuit board modules. The GVSC chip set is implemented in a processor card configuration as a double-sided microwire board approximately 6 inches by 6 inches in size. One side of the board contains the VHSIC processor and interface chips, while the other side contains 128K of radiation-hardened 64K SRAMs. This board technology offers controlled impedance, low cross talk, high density, ease of repair, and a high degree of reliability/maintainability. Since the 1750A chip set is the essence of the GVSC, then it is worthwhile and instructive to further explore the 1750A architecture.

The Honeywell 1750A processor design is a generic design capable of meeting the stressing performance requirements of complex missions as well as less demanding applications. The full-up processor consists of two partitions--the three-chip CPU and the two-chip FPP. The primary purpose of the FPP is to increase floating point performance. However, in a system with reduced requirements, the CPU is capable of operating alone, executing the full 1750A instruction set at better than 1.5 MIPS throughput to the DAIS mix. The full 1750A chip set (5 chips) typically delivers 4.7 MIPS.

Each CPU chip is associated with a major processor function. Chip 1 is instruction pipeline and control, Chip 2 is the arithmetic/logic execution portion, and Chip 3 is primarily concerned with memory address preparation. Chip partitioning minimizes chip interconnects and maximizes perforrance. Each chip contains its own clock input and test maintenance bus interface. Each chip also contains its own microcode control ROM, although all control ROM addressing originates from Chip 1.

The FPP is partitioned into two chips. FPP Chip 1 handles floating point add and subtract, while FPP Chip 2 performs floating point multiply and divide and fixed point multiply.

The five VHSIC chips are contained in ceramic packages with kovar covers, providing a hermetically sealed cavity and complete environmental protection for the chips. In the current configuration, a maximum of 284 I/Os can be used, which is adequate for the VHSIC chips that require a maximum of 240. External package dimensions are less than 1 inch on a side, and accommodate a chip of up to 0.5 inches on a side.

Each 1750A chip set has the following characteristics.

	MIPS	Operating Characteristics
Nominal	5.3	5V, 27°C
Worst Case	3.7	4.5V, 125°C
Typical	4.7	5V, 60°C

Word Size - 16, 32, 48 Bits

Memory Addressing - Up to 8M words

Power - Processor 2.6 watts (.52) watts/chip)
128K Memory 12.0 watts

Memory Error Detect/Correct - Parity (Error Detection and Correction, EDAC)

- GVSC uses EDAC; 1/3 word size devoted to EDAC/Spares; 10-year mission life assumed.

The Honeywell 1750A chip set is a custom chip set (Class S screened) for use in spaceborne applications. Since the model deals with airborne applications as well, it would be overkill to use this 1750A chip set in airborne or ground mobile applications. The following 1750A chip set was used for airborne and ground mobile applications. Since Honeywell 1750A chip set costs were not available, this chip set was also used to estimate the cost of Honeywell's 1750 chip set.

The Performance Semiconductor 1750A chip set consists of three chips. The chips are manufactured using CMOS .8 micron technology. The 1750A Class B chip set consists of a CPU, Processor Interface Circuit, and Memory Management Unit. The MMU is needed only if 64K or more words are to be accessed. The chip set will be used in ground, ground mobile, and airborne applications.

	FY88 Price Quote	
1750A CPU	3500.	
PIC	1750.	
MMU	2450.	
1750A Chip Set	7700.	(Class B)

The Honeywell chip set reflects a Class S application. Its costs were estimated by arbitrarily multiplying the 1750A Class B price by three.

Honeywell 1750A Chip Set (5 chips) Estimated Price	7700 * 3 = 23,100.	(Class S)

Common Signal Processor

The Common Signal Processor (CSP) is a high-performance, modular signal and data processing system that can be configured for a wide range of applications. Unique processing capabilities support use in radar, sonar, electronic warfare, communications, navigation, digital mapping, image processing, automatic target recognition, data fusion, and specialized classified applications.

Developed under Air Force contract and IBM funding, CSP is the first programmable signal processor to fully utilize: 1.25 micron, state-of-the-art, Very High Speed Integrated Circuits (VHSIC); Ada programming; and data-flow graph technologies. The multiprocessor architecture and the modular, building-block design of the CSP make it possible to configure both low-end processing systems for small subsystem applications and high-end processing systems having advanced supercomputer capabilities.

CSP is an architecture rather than a single configuration. The CSP configuration is a multiprocessor consisting of a set of modules, the quantity and types of which are application-dependent. The CSP architecture is partitioned into a set of core modules, functional element modules, and power supply modules.

Functional element modules are the major signal processing workhorses of a CSP configuration. They are divided into four categories:

- Memory modules provide bulk storage within a CSP configuration.

- Processing element modules provide the programmable signal processing capability.

- High-speed I/O modules provide interfaces into and out of a CSP configuration for sensor data, sensor control, inter-CSP data exchange, display data, and high-speed application load of a CSP configuration.

- Processor modules can be included in a CSP configuration to perform specialized signal processing functions.

Support element modules provide functions ancillary to and in support of the main signal processing flow. They are divided into three categories:

- General-purpose computer modules host application command programs, which establish the processing modules in the CSP configuration, generate sensor control parameters, and communicate with higher level computers in the system architecture.

- System I/O modules provide communication paths to these higher level computers.

- A user console interface module is used in a laboratory environment for attachment of a user console to a CSP configuration for debug and test.

The floating point processing element (FPPE) provides both floating point and fixed point signal processing capability. It is rated at 125 million floating point operations per second based on performing FFT operations. The module contains two dual 8K 32-bit word local stores for overlapped execution and I/O via the data network.

The global memory modules provides storage for one million 32-bit words accessible at a 25 MHz rate. Error correction and detection corrects any single chip error within each of the four banks of storage on the module. The module provides addressing modes, which allow the local operating system to treat storage as either first-in/first-out (FIFO) queues or unstructured buffers.

The element supervisor unit (ESU) is the common functional element controller used throughout a CSP configuration. It hosts the ESU component of the local operating system (LOS), which provides all capability needed to operate and monitor a functional element. The ESU controls its set of functional elements (up to six) via a 16-bit element control bus (ECB).

The microprocessor on the ESU implements a Mil-Std-1705A instruction set architecture plus execute input/outputs (X1Os) to control ECB and EMB operations. The module contains 256K words of program storage plus an 8K word start-up ROM. The ESU is capable of executing 2.8 million instructions per second.

Three module types from the 1750A program are planned for use in CSP configurations. These are the 1750A central processing unit (CPU), dual Mil-Std-1553B bus interface module, and dual high-speed system bus interface module. In addition, a user console interface module is used to provide attachment of a user console for laboratory use in debug and application development.

The 1750A CPU provides approximately three MIPS of general-purpose computing capability and is used in a CSP configuration to host the subsystem manager (SSM) portion of the local operating system; it is also used as backup for the SSM, and as a SSM support processor (SSMSP). The SSM provides all central management of the assets of a CSP configuration. The SSM backup, if used, provides a replacement for the SSM in the event of its failure.

CSP Performance Overview

Throughput
Floating Point - 150 to 1800 MFLOPS (150 MFLOPS per FPPE)
Fixed Point - 600 to 7200 MOPS (600 MOPS per FPPE)
General Purpose - 5 to 50 MIPS
FPPE - Implemented with 16 VHSIC chips

Memory (4 megabytes per global memory)
Data - 4 to 128 Megabytes
Program - 1 to 10 Megabytes
Type - 256K * 4 DRAM, 40 ns
Word Size - 32 bit plus 8 bits for EDAC
Global Memory - 40 (256*4) DRAMs + 3 VLSI chips (EDAC, Interface, Address Generator)

CSP delivered to Air Force in early July 1988.

Architecture consisted of:

- 12 FPPEs
- 6 Global Memories
- 9 ESUs

Estimated weight - 93 lbs

Estimated power - 880 watts

5
FOPE MODEL DOCUMENTATION

The FOPE Model is a fairly straightforward and simple Lotus-based approach that uses an engineering build-up methodology* to estimate costs of mobile-based digital processors. The following section describes the model in detail. Section 6 presents two cost estimates.

The cost drivers for the FOPE represent a departure from the usual digital processing drivers such as power, weight, and volume. The advances in technology and processor performance promised by VHSIC has cast doubt on the use of weight, power, and volume cost estimating relationships (CERs). Until components using VHSIC technology become more prevalent and VHSIC technology becomes more mature, a data base that reflects these emerging technologies will be difficult to compile. Without a data base to analyze, a cost relationship driven by physical descriptors cannot be relied upon to explain VHSIC-based processing costs.

Electronics using mature IC technologies such as SSI and MSI have shown that weight, power, and size explain costs to a large degree. The same is not necessarily true for VHSIC; it must still be confirmed later in the life cycle of the VHSIC program.

There are alternative parameters other than physical descriptors available to the analyst. They are millions of floating point operations per second (MFLOPS), millions of instructions per second (MIPS), and amount of memory in millions of bytes (Megabytes). The use of these parameters allows the analyst to estimate processor cost when power, weight, and size are not known and throughput and memory requirements are known. Early in a program's history, physical descriptors can be quite uncertain. Depending on the type of technology involved, power and weight estimates can vary considerably between the time of the first cost estimates to when the processor is actually produced. Throughput and memory requirements tend to be estimated with greater certainty and

*See figure 5.1 for the block diagram of the model's methodology.

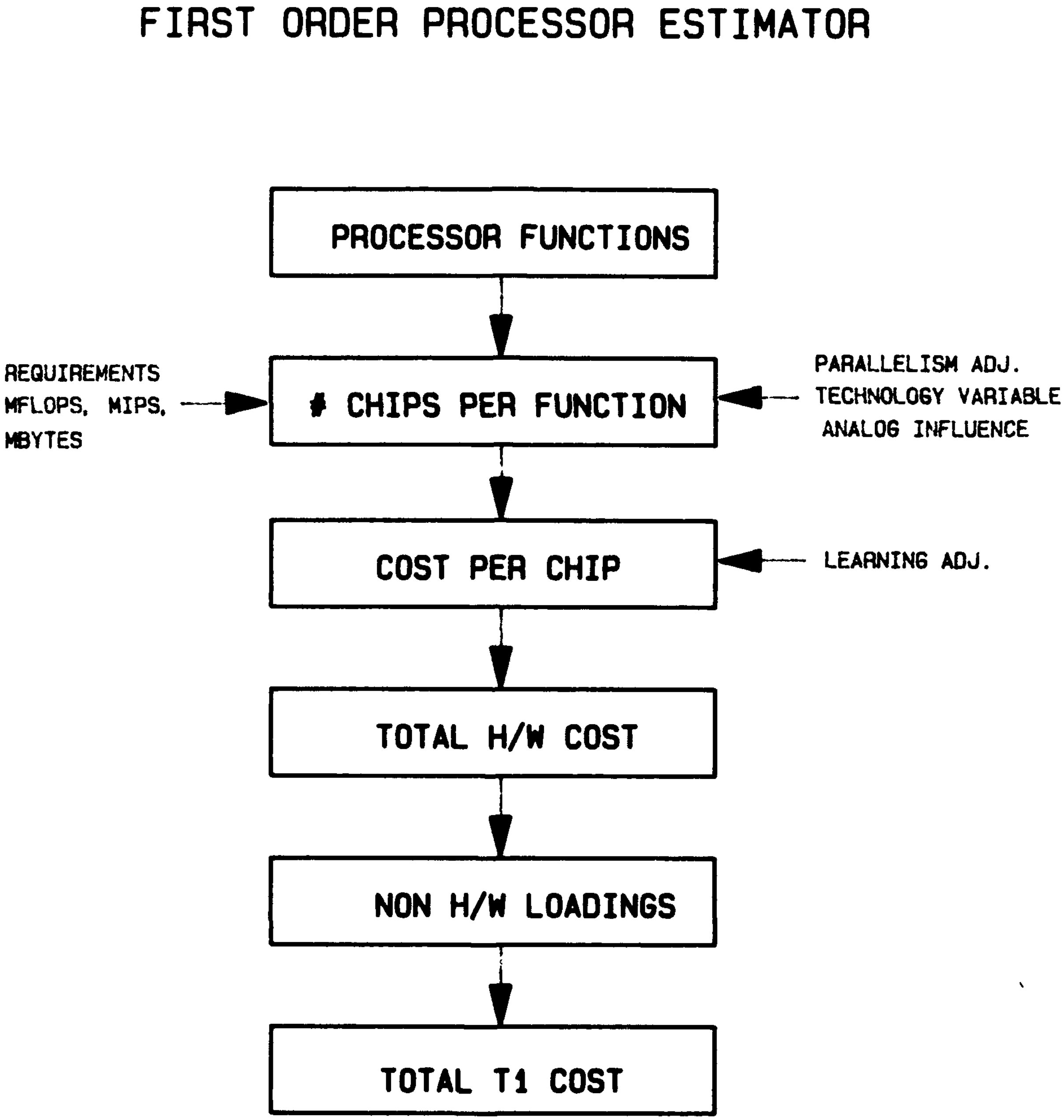


Figure 5.1. First order processor estimator.

stability, since they are dependent only on the type of problem to be solved and not the type of technology involved.

One last note on the model drivers. It is important not only to know the amount of memory but the type of memory. The model allows the user to specify percentages of several types of memory. They are:

Main Memory Category:	Dynamic RAM Static RAM EEPROM
Mass Memory Category:	Bubble Memory Disk

After inspection of figure 5.2, it is clear that costs can vary significantly depending on the allocation of memory type. This part of the model's input simply allows for memory hierarchies to be specified explicitly. This makes good sense, considering the potential cost consequences of differing memory hierarchies. To recap, the primary model drivers include:

	Function
Amount of Scalar Processing (MIPS)	Program Control
Amount of Vector Processing (MFLOPS)	Arithmetic
Amount of Memory (Mbytes)	Memory
Memory Hierarchy (% Allocations of Memory Type)	Memory

Model Inputs

System Name - Used for title and documentation purposes.

Base (A, GM, S) - The type of basing mode for the processor. This option modifies the cost of the H/W chips. The assumption is that airborne (A) and ground mobile (GM) use components processed to Mil-Std-883C, Method 5004, Class B. Space (S) based processors use components processed to Mil-M-38510, Class S. Class S components are assumed to cost 3 times more than Class B components.

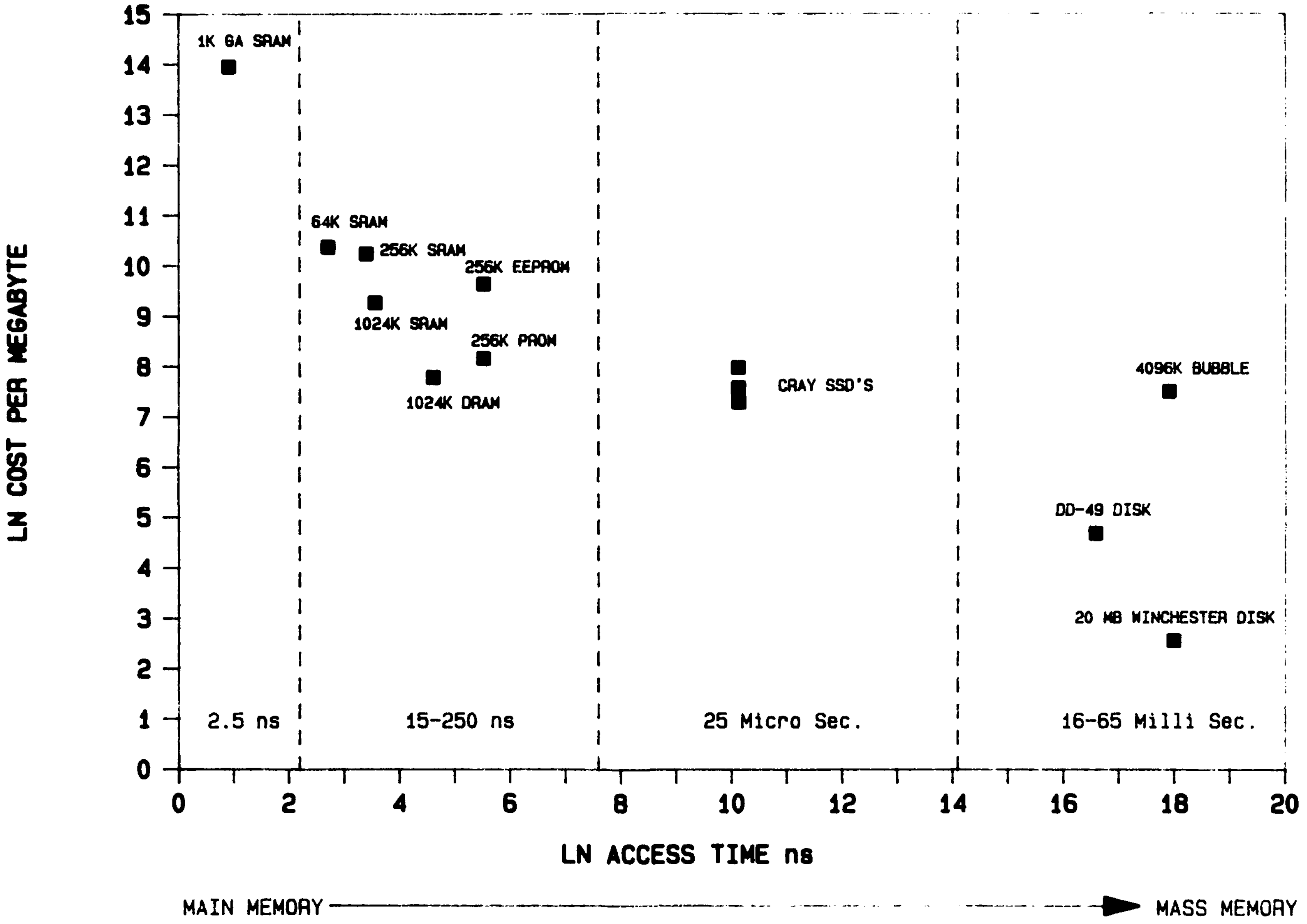


Figure 5.2. Memory devices, speed vs price.

> Author's Note: I talked at some length with Honeywell about Class S chips. Class S chips are being produced at a very low rate (i.e., you can hold all the Class S chips produced in the palm of your hand). Class S requirements are not consistent with VHSIC technology. For example, a Class S requirement of destructive testing is unrealistic, given low yields. The current thinking of qualifying single parts (part qualified) is giving way to the idea of qualifying and certifying design rules (line qualified). Thus, parts that come off a qualified line could then be screened Class B or Class S. Honeywell also stated that the 3 cost factor from Class B to Class S reflected reformed Class S certification procedures (line qualified) and that current Class S requirements could push the factor significantly higher.

Include Enclosure? - This input requires a "Y" or "N" answer. This appears in the model because spaceborne processor costs usually do not include enclosure costs, whereas in ground mobile and airborne platforms, enclosure costs are generally included.

Parallelism (A or R)? - This option refers to the efficiency of parallel processing. It modifies the number of processors needed in parallel to achieve the desired throughput. The parallelism question must be answered twice, once for scalar processing and next for vector processing. This allows for greater flexibility as parallel processing efficiencies for scalar processing can be quite different than that of vector processing. An answer of "A" invokes the Amdahl conjecture for parallel processing, while "R" invokes an 11:1 ratio rule. These two concepts will be discussed in the following text.

Parallel processing is most effective when the following items are present:

1. The problem to be solved can be broken into n independent similar tasks (for an n processor system), and be integrated back together to yield reasonable answers.

2. No or low operating system overhead effects.

3. No conflicts over memory access of communications paths.

If these hold, then speedup performance is linear with number of processors. This is known as "ideal" parallelism efficiency. However, reality dictates that ideal situations are rare indeed. There have been situations where researchers are experiencing close to ideal speedups in highly parallel processing systems (4096 processors linked, for example). However, to experience such high speedups, the problems and computing environment are usually contrived and limited in scope. Item 2 doesn't usually happen because as more processors are added, more controllers, buses, etc., are needed, and usually a more complicated operating system is needed. Item 3 is usually a processor architecture issue, and of course the whole parallel processing issue is moot if the problem/algorithm cannot be "parallelized." An important note is that parallel architectures that are given scalar problems (i.e., zero % "vectorizable") will yield the same performance as a uniprocessor system.

The FOPE model considers two upper bounds to deal with the issue of parallelism and its relationship with speedup over a single processor. The first is Amdahl's conjecture. Amdahl states that if a computer has two speeds of operation, the slower mode will dominate even if the faster mode is infinitely fast. This led to an upper bound estimate of:

$$\text{Speedup} = N/\ln(N)$$

where N is the number of Processors in Parallel. Thus, if 47 mips are required and each processor delivers 4.7 mips, then 10 processors are needed under ideal conditions. Amdahl would state that more than 10 processors are needed because:

$$10/(\ln 10) = 4.3 \text{ effective processors}$$

Therefore, 10 processors is effectively only 4.3 processors because of the additional overhead. 4.3 processors would yield only 20 mips, and not the 47 that are actually needed. This relationship is approximated by the following formula:

$$X = (2.21K^{1.25}) - (5.93K^{-.43})$$

where K = Number of processors in ideal state

X = Number of processors needed, assuming Amdahl state

Thus, to achieve 47 MIPS, K = 10 and the number of processors needed to achieve 47 MIPS is 37.

The other assumption is a ratio rule. The rule states that for every 11 processors, an additional processor is needed just to manage overhead (11:1). To invoke this assumption, answer "R"; otherwise, "A" will assume the Amdahl assumption. The ideal assumption is not addressed, but could be invoked by modifying the spreadsheet.

Learning Slope - The learning slope in % is entered. Cumulative average costs (CAC) are calculated for each type of chip. Lot sizes for IC chips are typically produced in quantities from 12 (one wafer) to 5000. Therefore, for chip quantities of 12 or less, a CAC_{12} is calculated, and for quantities greater than 5000, CAC_{5000} costs are calculated (i.e., no learning after 5000 units).

MIPS, MFLOPS, BYTES - The next set of inputs have to do with the previously discussed primary model drivers.

Input Costs - Known chip costs are entered. The input chip cost must reflect as close to a CAC_{12} (chip cost at $q<12$) cost as possible. The following chip costs reflect catalog and/or telephone quoted prices (FY88).

Actual chip prices that Honeywell and IBM use in the GVSC and CSP were not made available. Therefore, the next best course was to obtain information on chips from other sources that matched functionally with what was used in the two processors.

Performance Semiconductor, Inc. sells a Class B 1750A VHSIC chip set which includes the following chips (FY88 price):

1. CPU - $3500
2. Processor Interface Unit - $2450
3. Memory Management Unit - $1750

Total 1750A chip set cost - $7700 (Class B)
Performance - 4.7 MIPS (nominal)

The Class S functional equivalent is assumed to cost three times the Class B cost. Thus, Class S 1750A chip set costs $23100.

The arithmetic function has been modeled after the Common Signal Processor's Floating Point Processing Element. The FPPE contains many different ICs. However, the bulk of the IC componentry cost is reflected by the 16 VHSIC chips contained in the FPPE. Each FPPE delivers a peak rate of 150 MFLOPS. The following is a more detailed listing of each VHSIC chip. The list was extracted from a 1986 IBM Technical Interim Report on the CSP.

FPPE VHSIC Chips	Quantity
Data Network Switch	3
ROM Controller	1
Floating Point Memory Controller	6
Floating Point Function Generator	1
Floating Point ALU	2
Floating Point Controller	1
Floating Point Mult/Accum	2
	16

Performance: 150 MFLOPS (Peak)

The average cost for each VHSIC (Class B) chip is assumed to be $1000 ($CAC_{12}$). Class S VHSIC chips are assumed to be $3000 ($CAC_{12}$). Therefore, a Class B FPPE chip set is $16,000 and a Class S equivalent is $48,000.

Memory Chips: Main Memory

Static RAMS

Size	Speed	CAC_{12} Catalog Class B Price	CAC_{12} Estimated Class S Price	Source
64K*1	20 ns	$255	$ 765	Performance Semiconductor
256K*1	30 ns	$875	$2625	Performance Semiconductor

Dynamic RAM

Size	Speed	CAC_{12} Catalog Class B Price	CAC_{12} Estimated Class S Price	Source
1024K*1	100 ns	$300	--[*]	Micron, Boise, Idaho

EEPROMS

Size	Speed	CAC_{12} Catalog Class B Price	CAC_{12} Estimated Class S Price	Source
256K*1	250 ns	$480	$1440	SEEQ, San Jose, California

Memory Chips: Mass Memory

Disk	Size	Speed	CAC_{12} Off-the Shelf $/Mb	Source
CRAY SSD-7	4096 Mb	25 microsec	$1,465	Cray
DD-40	1200 Mb	16 millisec	$ 108	Cray

The disks listed here are high-performance mass storage devices for use on fixed platforms. A ruggedized variant conceivably could be used for ground mobile or airborne applications. Costs are in terms of dollars per megabyte.

[*] Since dynamic RAMS cannot be radiation hardened, Class S costs are not considered.

Bubble Memory

Size	Speed	Operating Temperature °C	CAC_{12}	Source
1024K	50 millisec	-20 +85	$1384	Mem Tech
4096K	50 millisec	0 +75	$ 907	Mem Tech
4096K	50 millisec	-40 +85	$3165	Mem Tech

Mem Tech, Inc. builds bubble memory for industrial, commercial, aerospace, and military applications. The $907 part is assumed to be for ground and airborne applications, while the $3165 part represents spaceborne applications.

The costs listed reflect catalog prices as of September 1988. Since IC cost is the central driver of processor costs, it is worthwhile to note the important cost drivers that affect IC cost.

From discussions with several chip suppliers and producers, the following points seem to drive chip yield and thus cost.

1. Level of Integration
2. Feature Size
3. Number of Layers or Masks
4. Process Type (Bipolar, MOS, GaAs, etc.)
5. Die Size, Wafer Size
6. Type of Packaging, Number of Pins
7. Number of Sources, Quantity Demanded
8. Custom and Semi-custom vs Standard ICs

Level of integration and feature size affect yield because as the circuitry becomes smaller there is a greater chance of defect (foreign objects) in the wafer area. Also, with more steps in the chip process, the chances that a defect will appear are increased. These defects are typically minimized by advances in clean room technology. Clean room technology, however, is quickly reaching technological limits. Advances may come about by processing chips in a robotic environment or vacuum which eliminates the dirtiest part of the process, the human interface.

The process type affects the density of transistors on the chip and how the chip is manufactured. The production of GaAs ICs, for example, is affected by gravity.

Die size and wafer size are directly related to the number of dies per wafer. Simply put, number of dies per wafer is generally positively correlated with yield.

The type of packaging can significantly affect chip (packaged) costs. Dual in line (DIP) packaging represents old technology and typically only amounts to 10 percent of chip (unpackaged) cost. To accommodate the increasing number of pinouts, manufacturers have to resort to newer packaging technologies. The newer packages include pin grid arrays, flatpacks, and multichip carriers. These packages typically use ceramic substrate (for heat dissipation), gold wire bonding, and very fine pin diameters. These newer packages could be as high as 50 to 60 percent of unpackaged chip cost.

The number of sources affects market pricing, and to some extent technical maturity. Quantity demanded is a big driver in chip cost. Apparently, yields increase as the number of producctions lots is increased. This is similar to a learning curve effect. Quantity demanded largely explains the reason why memory chips are generally cheaper than logic chips, even though the two are produced in a similar manner.

Obviously, custom ICs are more expensive than standard ICs. Testing, operating, performance, and environmental requirements seem to be the explanation for this increase (in both production and design). It is both curious and worth comment that non-VHSIC contractors are building "standard" VHSIC chips that perform at or above military requirements.

Figure 5.3 summarizes the default costs used in the FOPE model.

Performance - There are two performance variables--one each for the scalar and vector processing functions. The two parameters have been set to the GVSC and CSP throughputs.

		Source	
MIPS/1750A Chip Set	4.7	Typical Case	GVSC
MFLOPS/FPPE	150	Peak Case	CSP
VHSIC Chips/FPPE	16		CSP

		(Q < 12)	
		Ground Mobile,	
		Airborne	Level of
		Class B Screened	Integration
(Scalar)	1750A Chip Set (5)	$7700	VHSIC
(Vector)	FPPE Chip Set (16)	$16000	VHSIC
Main Memory	64K SRAM	$255	VHSIC
	256K SRAM	$875	VHSIC
	1024K DRAM	$300	VLSI
	256K EEProm	$480	VLSI
	256K Prom	$110	VLSI
Mass Memory	4096K Bubble Memory	$907	-
	Hi Performance Disk	$108/mb	-
	CRAY Solid State Disk	$1465/mb	-

- Class S screened parts assumed to be 3 times the cost of Class B.

  Class S increases due to:

  Immature Technology/Manufacturing Processes
  Radiation Hardening
  Destructive Testing
  Reliability Requirements
  Lower Demand

  Lower Yields

Figure 5.3. Summary costs.

6
MODEL OUTPUTS AND ESTIMATES

Memory Chips - Depending on the amount of memory required and the size of each chip, a quantity of chips is calculated that would meet the required memory size. There is an additional number of SRAMS and DRAMS added to account for EDAC and spares. The additional number uses the same scaling present in GVSC (SRAMS) and CSP (DRAMS).

Processing Chips - The number of processing chips is directly related to the throughput performance variables and the type of overhead allocation that is chosen (Amdahl or 11:1 overhead ratio). The overhead allocation only affects the number of processing chip sets (1750A and FPPE). The following ratio applies for chip quantity calculations uncorrected for increases in overhead due to parallelism.

# 1750A chips = (MIPS/4.7) * 5

# FPPE chips = (MFLOPS/150) * 16

# Airborne 1750A chip sets = # 1750 chips/3

# GVSC 1750A chip sets = # 1750 chips/5

# FPPE chip sets = # FPPE chips/16

Memory, Processing Cards - Scaled off of the CSP

# Memory Cards = # Memory Chips/43

# Processing Cards = # Processing Chips/16

CAC Cost - Cumulative average costs for each of the processing functions is calculated by the following formula.

$CAC(q) = A\ q^b$

where $A = CAC_{12}$ chip cost

CAC_{12} is used because the chip prices that were collected reflect quantities of 12 or less.

Cost per Function - Total hardware costs are a result of the multiplication of chip quantities and their respective CAC(q) costs. The hardware WBS reflects the functional elements of a generic processor. Note that another electronics line item appears, even though no chip quantity or price estimates were made for that function. Other electronics cost is factored as 50 percent factor of the CPU hardware cost. Other electronics consist of data, control, and address buses, drivers, and I/O interface chips.

Non-hardware Factors - Material handling, factory O/H, G&A, fee, labor, and structure are added to the hardware cost to arrive at a full processor cost.

	%	Source
Material Handling	10% of Hardware total	CR-210 - "Milstar Low Volume Force Element and Small Portable Terminal"
Factory O/H (Includes Manuf. Support, Sust. Eng., and recurring tooling)	50.8% of Cumulative Total	CR-210
G&A	13% of Cumulative Total	CR-210
Fee	12% of Cumulative Total	CR-210
Labor	11% of Loaded Mat'l	CR-273 - "Avionics Reliability Cost Impact Study"
Enclosure	$5000 * # (Slots)	Covers box, backplane, backplane interconnect assembly, connectors, and power conditioner*

Labor cost includes printed circuit board costs.

To see the model in action, the following pages document two estimates (GVSC and CSP).

*This is not the main power supply. Main power supply is not estimated by this model.

Estimate #1

GVSC:

Spaceborne (Class S screened ICs)

12 MIPS

0 MFLOPS

8 Megabytes

100% Main Memory

100% DRAMS

Tl 1988 Estimated Price: 1.7 Million

FIRST ORDER PROCESSOR ESTIMATOR

SYSTEM NAME	GVSC	
BASE (A,GM,S)	S	
INCL ENCLOSURE?	Y	
SCALAR PARAL. A OR R?	A	
VECTOR PARAL. A OR R?	R	
LEARNING		0.95

MIPS (SCALAR PROC.)	12
MFLOPS (VECTOR PROC.)	0
MBYTES (EDAC INCLD)	8
% MASS MEMORY	0%
% BUBBLE MEMORY	0%
% DISK DRIVE	0%
% MAIN MEM	100%
% SRAM MEM	100%
% DRAM MEM	0%
% EEPROM MEM	0%

CAC12 COSTS	
1750A $/CHIP SET	$23,100
FPPE $/CHIP SET	$48,000
$/SRAM	$765
$/DRAM	$900
$/EEPROM	$1,410
$/BUBBLE	$3,165
DISK ($/MB)	$108.00

PERFORMANCE	
MIPS/1750A CHIP SET	4.7
MFLOPS/FPPE	150
CHIPS/FPPE	16

SCALAR NODES	
O/H 1 PER 11	3
(AMDAHL) NODES	3

VECTOR NODES	
O/H 1 PER 11	0
(AMDAHL) NODES	0

T1 88$M = 1.701

	K BITS		BITS
SRAM MEM	64	X	1
DRAM MEM	256	X	4
EEPROM MEM	32	X	8
BUBBLE MEM	4096	X	1

MEMORY CHIPS	
# SRAMS	1024
# DRAMS	0
# EEPROMS	0
# BUBBLES	0
TOTAL	1024

PROCESSING CHIPS	
# FPPE CHIPS (VECTOR)	0
# MIP CHIPS (SCALAR)	15
TOTAL	15

TOTAL CHIP COUNT	1039
EST MEM CARDS	26
EST PROC CARDS	3

QUANTITIES			CAC COST		
1750A SETS	3	CAC	12	=	$23,100
#FPP SETS	0	CAC	12	=	$0
# SRAMS	1024	CAC	1024	=	$551
# DRAMS	0	CAC	0	=	$0
# EEPROMS	0	CAC	0	=	$0
# BUBBLES	0	CAC	0	=	$0

```
COST PER FUNCTION                                  PEAK POWER
CPU                                     0.069
  1750A (SCALAR PROC.)      0.069                     7.8     WATTS
  FPPE  (VECTOR PROC.)      0.000                     0.0     WATTS
OTHER ELECTRONICS                       0.035
MAIN MEMORY                             0.564
  STATIC RAMS               0.564                   256.0     WATTS
  DYNAMIC RAMS              0.000                     0.0     WATTS
  EEPROMS                   0.000                     0.0     WATTS
MASS MEMORY                             0.000
  BUBBLE COST               0.000                     0.0     WATTS
  DISK                      0.000

      TOTAL H/W 88$M                    0.668       263.8     WATTS


T1 H/W COST 88$M                        0.668
MAT'L HANDLING        10%               0.067
FACTORY O/H           50.8%             0.373
G & A                 13%               0.144
FEE                   12%               0.150
                                    ==========
LOADED MAT'L PRICE                      1.402
LOADED LABOR 11%                        0.154   INCLUDES PCB COST
                                    ==========
TOTAL PRICE 88$M T1                     1.556   W/O ENCLOSURE
LOADED ENCLOSURE                        0.145
TOTAL ASSEMBLY 88$M T1                  1.701
```

Estimate #2

CSP:

Airborne (Class B screened ICs)

20 MIPS

1800 MFLOPS

64 Megabytes

100% Main Memory

100% DRAMS

Tl 1988 Estimated Price: **1.48 Million**

```
FIRST                     ORDER         PROCESSOR  ESTIMATOR

SYSTEM NAME               CSP           T1 88$M =      1.479

BASE (A,GM,S)             A                       K BITS     BITS
INCL ENCLOSURE?           Y             SRAM MEM      256 X       1
SCALAR PARAL. A OR R? A                 DRAM MEM      256 X       4
VECTOR PARAL. A OR R? R                 EEPROM MEM     32 X       8
LEARNING                        0.95    BUBBLE MEM   4096 X       1

MIPS (SCALAR PROC.)               20
MFLOPS (VECTOR PROC.)           1800          MEMORY CHIPS
MBYTES (EDAC INCLD)               64    # SRAMS                   0
   % MASS MEMORY                  0%    # DRAMS                 512
     % BUBBLE MEMORY              0%    # EEPROMS                 0
     % DISK DRIVE                 0%    # BUBBLES                 0
                                                  TOTAL         512
   % MAIN MEM                   100%
     % SRAM MEM                   0%
     % DRAM MEM                 100%
     % EEPROM MEM                 0%        PROCESSING CHIPS
                                        # FPPE CHIPS (VECTOR)   224
CAC12 COSTS                             # MIP CHIPS (SCALAR)     30
1750A $/CHIP SET              $7,700              TOTAL         254
FPPE $/CHIP SET              $16,000
$/SRAM                          $875    TOTAL CHIP COUNT        766
$/DRAM                          $300
$/EEPROM                        $470    EST MEM CARDS            13
$/BUBBLE                        $907    EST PROC CARDS           20
DISK ($/MB)                  $108.00
                                        QUANTITIES          CAC COST
PERFORMANCE
                                        1750A SETS       10 CAC   12  =   $7,700
MIPS/1750A CHIP SET              4.7    #FPP SETS        14 CAC   14  =  $15,819
MFLOPS/FPPE                      150    # SRAMS           0 CAC    0  =       $0
CHIPS/FPPE                        16    # DRAMS         512 CAC  512  =     $227
                                        # EEPROMS         0 CAC    0  =       $0
SCALAR NODES                            # BUBBLES         0 CAC    0  =       $0
O/H 1 PER 11                       4
(AMDAHL) NODES                    10

VECTOR NODES
O/H 1 PER 11                      14
(AMDAHL) NODES                    47
```

```
COST PER FUNCTION                                                PEAK POWER
CPU                                                   0.298
  1750A (SCALAR PROC.)          0.077                              26.0    WATTS
  FPPE  (VECTOR PROC.)          0.221                              38.3    WATTS
OTHER ELECTRONICS                                     0.149
MAIN MEMORY                                           0.116
  STATIC RAMS                   0.000                               0.0    WATTS
  DYNAMIC RAMS                  0.116                             285.8    WATTS
  EEPROMS                       0.000                               0.0    WATTS
MASS MEMORY                                           0.000
  BUBBLE COST                   0.000                               0.0    WATTS
  DISK                          0.000

      TOTAL H/W 88$M                                  0.564       350.0    WATTS


T1 H/W COST 88$M                                      0.564
MAT'L HANDLING             10%                        0.056
FACTORY O/H                50.8%                      0.315
G & A                      13%                        0.122
FEE                        12%                        0.127
                                                 ==========
LOADED MAT'L PRICE                                    1.184
LOADED LABOR 11%                                      0.130   INCLUDES PCB COST
                                                 ==========
TOTAL PRICE 88$M T1                                   1.314   W/O ENCLOSURE
LOADED ENCLOSURE                                      0.165
TOTAL ASSEMBLY 88$M T1                                1.479
```

APPENDIX A

The general purpose processing data base consists of off-the-shelf, commercially available processing suites. Since technical and cost data were readily available, a cost estimating relationship type methodology was used to relate cost as a function of a processor's cycle time, memory, and power consumed. Definition of each appears below.

Price - Catalog and telephone quote in millions of FY87 dollars.

Cycle/access time, nanoseconds indicates two benchmarks of the system's main storage. The cycle time is a minimum time interval that must elapse between the starts of two successive accesses to any one storage location. Though cycle time ranks with word length as one of the most significant individual indicators of a computer's performance potential, one cannot assume that the computer with the fastest cycle time will be the best overall performer in a particular application. Other parameters that have an important effect on a computer's performance include the flexibility and power of its instruction repertoire, the number of storage cycles it requires to execute each instruction, and its input/output capabilities. Access time is the actual elapsed time between the CPU's request for data and the time when that data is received (read) in memory.

Memory - Amount of main memory in megabytes (millions of bytes).

Power - Measured in kilowatts. Total power consumed by processor including cooling.

The range of the data base is as follows:

Price Cost:	.592M$ → 21.74 M$
Cycle Time:	4.1 ns → 45 ns
Memory:	32 mb → 2304 mb
Power:	3.7 kW → 360 kW

Total Data Points	51
Excluded Data Points	8

Eight data points were excluded from the original 51. The Amdahl group was excluded because the power estimates for those machines (after analysis) seemed much too low. Three other data points were excluded because they represented the "low end" of the various product lines. The prices quoted for these "low end" machines (Cray XMP/14, IBM 3090-120,-150) may be discounted because the equation overestimated these machines' costs by an average of 160 percent.

Regression Results

Price 87$M =	.502	$(\text{Mb Memory})^{.129}$	$(\text{Cycle Time})^{-.288}$	$(\text{Power})^{.592}$
t-stats	(-1.70)	(3.64)	(-3.58)	(12.89)

Standard Error (Log Space)	=	.2316
R^2	=	.951
Adjusted R^2	=	.947
Degrees of Freedom	=	39

The data base appears on the next page.

GENERAL PURPOSE GROUND BASED COMPUTERS

		MEMORY MB	CYCLE TIME	POWER (KW)	COST M$87
1	AMDAHL 5890-200	64	15.0	45.2	3.8
2	AMDAHL 5890-600	128	15.0	90.0	8.5
●3	AMDAHL 5990-700	64	10	22.43	6.733
●4	AMDAHL 5990-700	128	10	22.72	7.74
●5	AMDAHL 5990-700	256	10	23	8.77
●6	AMDAHL 5890-1400	128	10	45.13	12.51
●7	AMDAHL 5890-1400	512	10	45.7	15.6
8	AMDAHL 1400	128	5.5	54	6.5
9	AMDAHL 1200	64	5.5	42	5.5
10	AMDAHL 1100	32	5.5	33	4.0
11	ETA CYBER 205-600	128	20.0	126.0	5.7
12	ETA 10-P (1 CPU)	64	24.0	8	.81
13	ETA 10-P (1 CPU)	128	24.0	8.1	.952
14	ETA 10-P (1 CPU)	256	24.0	8.2	1.286
15	ETA 10-P (1 CPU)	512	24.0	8.3	1.762
16	ETA 10-Q (1 CPU)	64	19.0	9.6	1.143
17	ETA 10-Q (1 CPU)	128	19.0	9.72	1.33
18	ETA 10-Q (1 CPU)	256	19.0	9.84	1.619
19	ETA 10-Q (1 CPU)	512	19.0	9.96	2.095
20	ETA 10-E (1 CPU)	256	10.5	30.0	5.238
21	ETA 10-E (1 CPU)	512	10.5	30.375	6.048
22	ETA 10-E (1 CPU)	1024	10.5	30.775	7.667
23	ETA 10-G (2 CPU)	512	7.0	90.0	8.524
24	ETA 10-G (2 CPU)	1024	7.0	91.125	10.143
25	ETA 10-G (2 CPU)	2048	7.0	92.264	13.524
26	ETA 10-G (8 CPU)	2048	7.0	360.0	21.738
●27	CRAY XMP/14SE	32	8.5	70.66	2.38
28	CRAY XMP/EA116	128	8.5	110.29	5.8
29	CRAY XMP/EA216	128	8.5	126.56	7.8
30	CRAY XMP/EA232	256	8.5	126.56	8.1
31	CRAY XMP/EA264	512	8.5	143.02	9.52
32	CRAY XMP/416	128	8.5	197.74	11.19
33	CRAY XMP/432	256	8.5	197.74	11.90
34	CRAY XMP/464	512	8.5	209.08	13.33
35	CRAY 2S/2-64	512	4.1	94.67	11.43
36	CRAY 2S/2-128	1024	4.1	123.23	14.76
37	CRAY 2S/4-128	1024	4.1	172.23	16.67
38	CRAY 2D/4-256	2048	4.1	210.46	14.76
39	CRAY YMP/8	2304	6.0	272.55	19.05
●40	IBM 3090 - 120	32	18.5	26.88	.681
●41	IBM 3090 - 150	32	17.75	28.54	1.19
42	IBM 3090 - 180	32	17.2	28.97	2.095
43	IBM 3090 - 200	64	17.2	37.67	3.905
44	IBM 3090 - 300	64	17.2	44.46	5.33
45	IBM 3090 - 400	128	17.2	71.69	7.45
46	IBM 3090 - 600	128	17.2	85.26	9.851
47	VAX8810	48	45	3.7	.592
48	VAX8820	128	45	8.3	.885
49	VAX8830	128	45	9.9	1.162
50	VAX8840	128	45	11.5	1.473
51	VAX8842	256	45	16.3	1.694

● Excluded from regression analysis.

```
          *********** Percentage  Error Table ***********
 ---------------------------------------------------------------------
 |     Observations      Actuals     Predicted     Residuals     % Errors|
 |     --------------------------------------------------------------------|
 |1  AMDAHL 5890-200        3.80          3.76          0.04        -0.99|
 |2  AMDAHL 5890-600        8.50          6.19          2.31       -27.23|
 |3  AMDAHL 5990-700        6.73          2.79          3.94       -58.53|
 |4  AMDAHL 5990-700        7.74          3.08          4.66       -60.24|
 |5  AMDAHL 5990-700        8.77          3.39          5.38       -61.35|
 |6  AMDAHL 5890-1400      12.51          4.62          7.89       -63.07|
 |7  AMDAHL 5890-1400      15.60          5.57         10.03       -64.32|
 |8  AMDAHL 1400            6.50          6.10          0.40        -6.14|
 |9  AMDAHL 1200            5.50          4.81          0.69       -12.59|
 |10 AMDAHL 1100            4.00          3.81          0.19        -4.72|
 |11 ETA CYBER 205-600      5.70          6.95         -1.25        21.91|
 |12 ETA 10-P (1 CPU)       0.81          1.18         -0.37        45.59|
 |13 ETA 10-P (1 CPU)       0.95          1.30         -0.35        36.47|
 |14 ETA 10-P (1 CPU)       1.29          1.43         -0.15        11.29|
 |15 ETA 10-P (1 CPU)       1.76          1.58          0.19       -10.53|
 |16 ETA 10-Q (1 CPU)       1.14          1.40         -0.26        22.92|
 |17 ETA 10-Q (1 CPU)       1.33          1.55         -0.22        16.38|
 |18 ETA 10-Q (1 CPU)       1.62          1.71         -0.09         5.32|
 |19 ETA 10-Q (1 CPU)       2.10          1.88          0.22       -10.35|
 |20 ETA 10-E (1 CPU)       5.24          3.91          1.33       -25.32|
 |21 ETA 10-E (1 CPU)       6.05          4.31          1.74       -28.74|
 |22 ETA 10-E (1 CPU)       7.67          4.75          2.92       -38.04|
 |23 ETA 10-G (2 CPU)       8.52          9.21         -0.69         8.07|
 |24 ETA 10-G (2 CPU)      10.14         10.15         -0.01         0.05|
 |25 ETA 10-G (2 CPU)      13.52         11.18          2.34       -17.33|
 |26 ETA 10-G (8 CPU)      21.74         25.03         -3.29        15.14|
 |27 CRAY XMP/14SE          2.38          5.28         -2.90       121.72|
 |28 CRAY XMP/EA116         5.80          8.21         -2.41        41.63|
 |29 CRAY XMP/EA216         7.80          8.91         -1.11        14.25|
 |30 CRAY XMP/EA232         8.10          9.75         -1.65        20.32|
 |31 CRAY XMP/EA264         9.52         11.46         -1.94        20.37|
 |32 CRAY XMP/416          11.19         11.61         -0.42         3.71|
 |33 CRAY XMP/432          11.90         12.69         -0.79         6.66|
 |34 CRAY XMP/464          13.33         14.35         -1.02         7.63|
 |35 CRAY 2S/2-64          11.43         11.07          0.36        -3.15|
 |36 CRAY 2S/2-128         14.76         14.15          0.61        -4.12|
 |37 CRAY 2S/4-128         16.67         17.25         -0.58         3.50|
 |38 CRAY 2D/4-256         14.76         21.25         -6.49        43.94|
 |39 CRAY YMP/8            19.05         22.53         -3.48        18.27|
 |40 IBM 3090 - 120         0.68          2.38         -1.70       249.65|
 |41 IBM 3090 - 150         1.19          2.50         -1.31       109.80|
 |42 IBM 3090 - 180         2.10          2.54         -0.45        21.32|
 |43 IBM 3090 - 200         3.90          3.25          0.66       -16.84|
 |44 IBM 3090 - 300         5.33          3.58          1.75       -32.80|
 |45 IBM 3090 - 400         7.45          5.20          2.25       -30.23|
 |46 IBM 3090 - 600         9.85          5.76          4.09       -41.54|
 |47 VAX8810                0.59          0.60         -0.01         1.49|
 |48 VAX8820                0.88          1.10         -0.22        24.30|
 |49 VAX8830                1.16          1.22         -0.06         5.08|
 |50 VAX8840                1.47          1.33          0.14        -9.42|
 |51 VAX8842                1.69          1.79         -0.10         5.90|
 ----------------------------------------------------------------------
  Avg (Arith)           6.90          7.02          -0.12         2.36%
  Avg (Absolute)                                     1.15        17.25%

 Average Actual (Avg Act) in Unit Space.................       6.90
 Standard Error (SE) in Unit Space......................       1.85
 Root Mean Square (RMS) of % Errors.....................      21.5%
 Mean Absolute Deviation (MAD) of % Errors..............      17.2%
 Coef of Variation based on Std Error(SE/Avg Act).......      26.9%
 Coef of Variation based on MAD Res (MAD Res/Avg Act)..       16.7%
```

BIBLIOGRAPHY

Valkenburgh, Van, "Basic Solid State Electronics," Hayden Book Co., New Jersey, 1978.

Desrochers, George R., "Principles of Parallel and Multiprocessing," McGraw-Hill, New York, 1978.

Ludeman, Lonnie C., "Fundamentals of Digital Signal Processing," Harper and Row, San Francisco, 1986.

Graf, Rudolph F., "Modern Dictionary of Electronics, 6th edition," Sams & Co., Indianapolis, 1982.

Zaks, Rodnay, and Wolfe, Alexander, "From Chips to Systems," Sybex, Berkeley, 1987.

Hurd, Michael R., "Handbook of Space Technology: Status and Projections," CRC Press, Boca Raton, 1985.